AD P000970

NUMERICAL GRID GENERATION
Joe F. Thompson, editor

Conformal Grid Generation

by

David C. Ives
United Technologies Corporation
Pratt & Whitney Aircraft Group
East Hartford, Connecticut 06108

ABSTRACT

This paper treats conformal mapping as it relates to the generation of grids to be used for flow simulation. Classical and contemporary mapping methods are discussed and compared in some detail. Numerous suggestions are included to help the reader avoid common pitfalls, and the mapping methods believed most promising are identified.

INTRODUCTION

Conformal mapping is a versatile component of the spectrum of grid generation techniques, yet its use is sometimes avoided by investigators who feel uneasy about its implementation due to unfamiliarity. The object of this paper is to put conformal mapping into perspective, to discuss alternate implementations, and to induce the reader to use conformal mapping, when appropriate, as one component of a grid generation process.

Although this paper discusses the generation of grids using conformal mapping techniques, the grids themselves are not restricted to being conformal or orthogonal. Algebraic techniques are considered for the latter stages of a mapping sequence, and elliptic techniques are considered for "filling-in" the interior of a grid once the boundary correspondences have been obtained through a mapping sequence. The emphasis is on grids on a surface, rather than throughout a volume, since a conformal mapping is basically a surface-to-surface correspondence.

This paper starts with a discussion of the basic differences between algebraic, orthogonal, and conformal coordinate systems. Some implications of these differences are considered as they relate to flow simulation and design considerations. A brief review of complex variable notation and various types of conformal mappings is followed by a discussion of the problem of multiple valuedness, which can be the most troublesome aspect of conformal mapping from a computer implementation viewpoint. Various mapping techniques in current use are reviewed and techniques useful for creating new conformal mappings are outlined. The generation of a grid, once the mapping is known, is discussed.

Suitable reference material is listed by problem type and packaged computational tools are described. The closing remarks highlight techniques that are believed to be the most promising in this evolving field.

COORDINATE TRANSFORMATIONS

This section illustrates some of the differences between general, independent algebraic, orthogonal, and conformal coordinate systems.

A general two-dimensional transformation between (x,y) physical coordinates and (ξ,η) computational coordinates (where (ξ,η) are taken to be orthogonal throughout this paper) can be written as

$$x = x(\xi,\eta)$$
$$y = y(\xi,\eta)$$

or in differential form in terms of a Jacobian matrix as

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} \tag{1}$$

where, from the chain rule, the four independent parameters a, b, c, and d are given by

$$a = \left.\frac{\partial x}{\partial \xi}\right|_\eta$$
$$b = \left.\frac{\partial x}{\partial \eta}\right|_\xi$$
$$c = \left.\frac{\partial y}{\partial \xi}\right|_\eta$$
$$d = \left.\frac{\partial y}{\partial \eta}\right|_\xi \quad .$$

Here dx and dy are components of a differential vector dZ in the physical plane, while $d\xi$ and $d\eta$ are orthogonal (perpendicular) components of a differential vector $d\zeta$ in a computational plane. For a general algebraic transformation, dx and dy are not orthogonal, and their magnitudes need not be equal when the magnitudes of $d\xi$ and $d\eta$ are equal. An elliptic technique as in Ref. [1], for example, falls into the above class if orthogonality is not enforced.

A less general algebraic transformation (referred to as an independent algebraic transformation) composed of independent transformations in each direction can be written as

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} h & 0 \\ 0 & g \end{bmatrix} \cdot \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} \tag{2}$$

where h and g are two independent parameters. For this transformation, dx and dy are orthogonal but their magnitudes are not equal when the magnitudes of $d\xi$ and $d\eta$ are equal.

An orthogonal differential coordinate transformation can be written in a similar manner as

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} h & 0 \\ 0 & g \end{bmatrix} \cdot \begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix} \cdot \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} \tag{3}$$

where h, g, and Θ are three independent parameters, Θ is the angle between the dx and $d\xi$ directions, and dx is orthogonal to dy.

A conformal differential coordinate transformation can be written as

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} h & 0 \\ 0 & h \end{bmatrix} \cdot \begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix} \cdot \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} \tag{4}$$

where h and Θ are two independent parameters. For this transformation, dx and dy are orthogonal and their magnitudes are equal when the magnitudes of $d\xi$ and $d\eta$ are equal. Thus the magnification h (called the metric or mapping modulus) between the (x,y) and (ξ,η) planes is not a function of direction; it is just a function of location. A conformal transformation is also angle preserving in a local sense, so that if two differential vectors meet at the angle α in the (ξ,η) plane, they also meet at the angle α in the (x,y) plane.

FLOW SIMULATION IMPLICATIONS

A grid can affect the storage and speed of a flow simulation. This section touches on some of the aspects of this dependence of a flow simulation on the grid.

The computer storage required for a grid was once an important consideration for flow simulation. For a finite difference solution, the independent parameters in the differential coordinate transforms above were usually stored (for an orthogonal grid) or regenerated as required (for a grid constructed using only two independent algebraic stretchings). For a limited set of flow problems, which fortunately includes two-dimensional transonic potential flow, it turns out that a conformal mapping requires the storage of only the metric, h, at each grid point in addition to the potential, ϕ. The x and y physical plane coordinates and the parameter Θ in Eq. (4) are not needed throughout the grid in this case, due to the way the gradient operator transforms under a conformal mapping, so only two quantities need be stored for each grid point.

This is one reason why conformal grids were popular in the early days of computers for finite difference solutions using a velocity potential. Purely algebraic grids were also popular as they could be regenerated as required, so that only the potential need be stored at each grid point. The rapidly declining cost of computer memory has made storage requirements a minor issue* at many installations for two-dimensional and axisymmetric inviscid flow problems; in fact, a modern finite volume technique may routinely store twenty to thirty quantities at each grid point. The storage requirements of a grid system therefore need not be a deciding factor in the choice of a grid system.

The computer time required by a flow algorithm depends on the nature of the grid system. For example, when a conformal grid (or an orthogonal grid using a conformal step followed by independent algebraic stretchings) is used, a byproduct is the solution for the two-dimensional incompressible flow as discussed later. This incompressible solution may often be used with a compressibility transformation as an accurate first guess for an iterative solution of the compressible problem, reducing the computer time needed to converge the iterative process. In addition, a suitable conformal mapping often automatically provides a <u>two-dimensional concentration</u> of the mesh in regions of high gradient (because the mapping modulus is independent of direction for a conformal mapping), without simultaneously producing a one-dimensional concentration where not needed. This allows the use of fewer grid points, which results in faster calculations.

DESIGN IMPLICATIONS

A number of modern aerodynamic design methods, Refs. [2,3], wherein the pressure is specified and the body shape giving this pressure is calculated, use conformal transformation techniques. The computer programs required for such a conformal mapping based aerodynamic design process are minor variations of the conformal mapping programs required to generate an orthogonal grid. For self-consistency, one might as well use the same routines for both. This may prove to be a significant justification for conformal grid generation.

COMPLEX VARIABLE NOTATION

Complex variable notation allows us to write the differential conformal transformation, Eq. (4), in a compact form as

---

* In some computing installations, the charging algorithm recognizes this fact so that the charge for a calculation only depends on the CPU time and not on the product of CPU time and storage.

$$dx + idy = he^{i\Theta}(d\xi + id\eta)$$

or
$$dZ = H\, d\zeta$$

or
$$Z = \int H\, d\zeta$$

or
$$Z = F(\zeta)$$

where
$$Z = x + iy$$

and
$$H = he^{i\Theta}$$

and
$$\zeta = \xi + i\eta \quad .$$

Thus given the relation $Z = F(\zeta)$ we can easily calculate the mapping modulus as $h = \left|\frac{dZ}{d\zeta}\right| = |F'(\zeta)|$. Complex notation greatly simplifies the use of conformal transformation techniques, and is well supported* on most modern computing systems.

CONFORMAL MAPPING

Conformal mapping provides a surface correspondence that is not limited to planar surfaces. Conformal mappings from non-planar surfaces, such as from spheres or general cylindrical surfaces, to a plane have been employed by cartographers under the title of projections. We now will discuss some characteristics of these different types of mappings, starting with some simple planar mappings.

Planar mappings, relating points in two different planes, are the most familiar. An important planar mapping is the bilinear transform given by

$$\frac{(Z-a)}{(Z-b)}\,(c-a) = \frac{(\zeta-A)}{(\zeta-B)}\,(C-A) \quad , \tag{5}$$

where the points Z = a, b, and c correspond to the points ζ = A, B, and C respectively. It can be shown that Eq. (5) transforms circles or straight lines in the Z plane to circles or straight lines in the ζ plane. A region at infinity can be treated as a point in this equation. This mapping illustrates that one may specify the correspondence of only three specific points between planes (i.e., one may assign a, b, c and A, B, C in an arbitrary manner). The bilinear transformation is often used after another conformal mapping step to produce a canonical (or standardized) contour for the next step.

---

* A breakup into real and imaginary parts, and then using real arithmetic only, can produce a two to one speed improvement on IBM systems. This means that inner loops for heavily used routines should be considered for such a breakup.

Often a contour is conformally mapped onto a unit circle and then the conformal transformation

$$\begin{aligned}\omega &= \ln(\zeta)\\ &= \ln(re^{i\Theta})\\ &= \ln r + i\Theta\end{aligned} \tag{6}$$

is applied, followed by the independent algebraic transformation

$$\begin{aligned}R &= e^{\ln r} = r\\ \Theta &= \Theta\end{aligned} \tag{7}$$

to map the interior of the unit circle to the interior of the rectangle given by

$$0 \le R \le 1 \quad ,$$

$$0 \le \Theta \le 2\pi \quad .$$

A uniform grid in this rectangle plane may then be constructed and mapped back to the physical plane where the resulting grid will be orthogonal.

A simple conformal mapping to map a general axisymmetric cylindrical surface onto a plane has been variously reported in the literature and is well recognized overseas, Refs. [4, 5, 6, 7, 8, 9], but evidently not at present in the United States. This mapping is given by

$$dZ = r\, d\zeta \quad , \tag{8}$$

where

$dZ = ds + ird\Theta \quad , \quad d\zeta = d\xi + id\eta \quad ,$ and

s = arc length along cylinder in axial direction $= \int [1 + \left(\frac{dr}{dx}\right)^2]^{\frac{1}{2}} dx$.

The modulus of this transformation is simply r, the radius in the cylindrical coordinate system (x,r,Θ) describing the physical surface. For a constant radius axisymmetric cylinder, the radius r drops out of the flow equations and one can think of simply unwrapping the cylinder, but in fact one is implicitly performing a conformal mapping. For an axisymmetric conical surface, the above relation becomes

$$Z = e^{\zeta \sin\phi} \quad , \tag{9}$$

where ϕ is the angle between the axis and the cone surface. One obvious use of cylindrical mappings is in the generation of grids for quasi three-dimensional blade-to-blade flow simulations for axial, mixed (meaning both significant radial and axial flow components), and centrifugal flow turbomachines. For $\phi = \pi/2$, Eq. (9) reduces to Eq. (6) which has been used in Ref. [10] to map a centrifugal impeller with near log-spiral blades onto a two-dimensional cascade of blades.

The requirements for constructing a useful geographical or astronomical map (i.e: local angles are preserved and the local scale factor does not depend on orientation) are identical to the definition of a conformal mapping. Thus, the mappings developed by cartographers to represent a spherical surface by a planar map apply directly. Among such maps are stereographic and Mercator projections, and a more general projection used for star maps, Ref. [9]. Stereographic projections, invented by the Greek astronomer Ptolemy as cited in Ref. [11], were used to develop grid systems for the computation of supersonic flow over conical bodies, Ref. [12], and more general bodies, Ref. [13]. A spiral groove spherical bearing lubrication study, Ref. [14], also used stereographic projection. Stereographic projections have been known for 18 centuries; the recent innovation was in how to use them for flow calculations.

RIEMANN SHEET DETERMINATION

The bane of conformal mapping is the possibility of getting on the wrong Riemann sheet, or equivalently, choosing an inappropriate root. Development of conformal mapping computer programs would be substantially easier were it not for the multivalued nature of many mappings. This section discusses techniques that have proven useful for determining the appropriate root.

To illustrate the problem, consider the mapping

$$Z = \zeta^k \tag{10a}$$

where k is a real number. We can express ζ in polar coordinates as

$$\zeta = re^{i(\Theta+2\pi n)} \quad , \tag{10b}$$

where n is any integer, since

$$e^{i2\pi n} = 1 \quad .$$

Then

$$Z = r^k e^{ik\Theta} e^{ik2\pi n} \quad .$$

For a square root transformation $k = 1/2$, and the $e^{ik2\pi n}$ term is 1 if n is even and -1 if n is odd so that there are two different solutions for Z for the same ζ (while there is only one value of ζ for each value of Z). This multivaluedness can be visualized in terms of Riemann sheets, with each different value of ζ lying on a different sheet as illustrated in Fig. 1. For $k = 1/3$, there are three values of Z for each value of ζ. For irrational values of k, there are an infinite number of values of Z for each value of ζ.

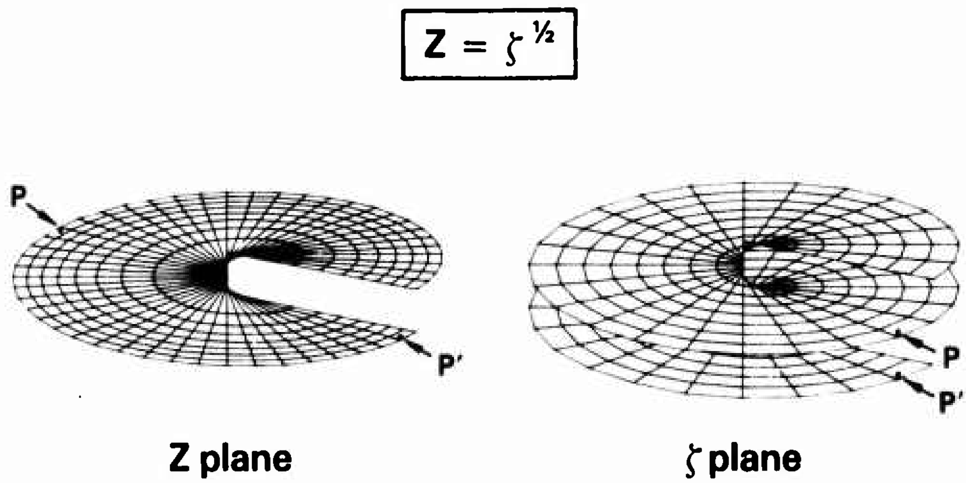

FIGURE 1. RIEMANN SHEETS

A computer implementation of $Z = \zeta^{1/2}$ using the CSQRT complex square root FORTRAN function will return only the root having a positive real part; the other root will be the negative of this root. It is up to the investigator to provide program logic to choose which of these two roots is appropriate.

For $k = 1/2$, which is rather common, physical reasoning can often be used to select the appropriate value of Z. In particular, mappings using the principle of reflection (covered later) produce quadratic equations with two roots which are image points with respect to a circle. In this case one may choose the root either inside or outside the circle based on physical reasoning. For the square root transformation, as often used in airfoil or wing grid generation, or the transformation,

$$Z = \zeta + e^{\zeta} \quad , \tag{11}$$

which also has two roots and is often used for inlet grid generation, the root selection is rather simple. Referring to Figs. 2 and 3, the initial value at "A" is the root with a negative real value, and as one traverses the airfoil in a clockwise direction and the inlet in a counterclockwise direction, the root to the right of, and (of those remaining, if any) nearest to, the previous root is chosen. This simple algorithm has worked well for a large variety of airfoil and inlet grid calculations.

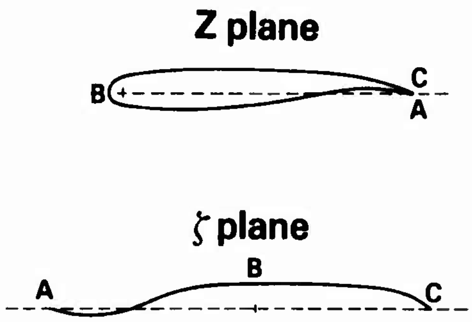

FIGURE 2. AIRFOIL MAPPING

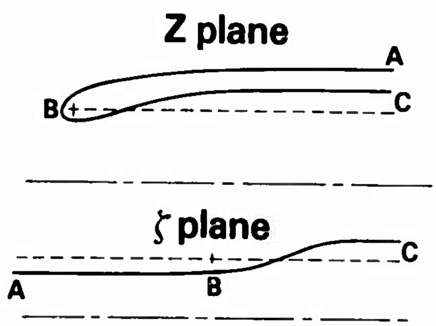

FIGURE 3. NACELLE MAPPING

For a transformation involving an irrational power of k near (but not exactly) 1/2, there are many values of Z that lie near each other and the selection is less obvious. The von Karman-Trefftz transformation, Ref. [15], given by

$$\frac{\zeta-a}{\zeta-b} = \left[\frac{Z-A}{Z-B}\right]^k \tag{12}$$

which transforms an airfoil in the Z plane into a near circle in the ζ plane, is one such case. When $k = \frac{1}{2}$, the trailing edge is cusped and Eq. (12) becomes the Joukowski transformation.

One technique to deal with the root problem is that of "tracking", where one chooses the value of n such that $(\Theta+2\pi n)$ in Eq. (10b) is closest to the value of $(\Theta+2\pi n)$ for a nearby point. The value of Θ is obtained from the real and imaginary parts of ζ using the FORTRAN ATAN2 function, which returns a value of Θ between $-\pi$ and π. In this manner one "tracks" the argument $(\Theta+2\pi n)$ continuously along a curve and chooses n to make this argument continuous. There remains the problem of how to choose the root for the initial point. This may often be accomplished from knowledge of the behavior of the transformation at the point in the mapped plane which corresponds to the far field in the physical plane.

Another method of root selection is to introduce a "branch cut" based on physical reasoning, and then preassign n on each side of the cut. The choice of root for a reflection mapping as described earlier is equivalent to using a circle for the cut. For many problems the cut may be taken as a straight line and is chosen so as to not cross the boundary being mapped. Some mappings may require more than one cut, as illustrated in Refs. [16] and [17]. Using a branch cut can be considered as a static technique to preassign n on a global basis, while tracking is a dynamic technique to determine n on a local basis.

It is not unusual, for complicated mappings, to occasionally encounter root selection errors in a conformal mapping program which had been thought to be resistant to such effects. This usually happens when attempting to map a geometry significantly different from, but still in the same class as, the geometry used to develop the program. For simple mappings (i.e: those quoted so far in this discussion with the possible exception of Eq. (12)) root selection problems are not expected if the above techniques have been carefully implemented. In any case, an error is easily detected by defining a uniform grid in the rectangular computational plane and mapping this grid back to the physical plane where it is plotted. If the physical plane grid is continuous, does not overlap, and does not exhibit open sections or jumps, the roots have probably been selected correctly. It is important to visually inspect the grid created for each new geometry before it is used for a flow calculation if there is any doubt on this point.

CLASSICAL MAPPING

The classical technique of mapping consists of mapping a contour (an airfoil, a cascade, an inlet, etc.) to a near-circle by a single transformation or a sequence of simple transformations. The near-circle is then mapped to a circle by a near-circle to circle transformation, such as proposed by Theodorsen and Garrick, Ref. [18].

The curve to near-circle transformation(s) often require a substantial level of ingenuity and luck, and even then can fall prey to root selection errors. In simple cases, such as for an isolated airfoil or an inlet, the near-circle is actually near to being a circle. For other cases (inlet with centerbody far inside or far outside, turbine cascade, small gap/chord compressor cascade), the near-circle may not even resemble a circle. When this happens, the investigator must either find a better mapping to a near-circle (which may not be easy in practice, even after many years of experience), or change to a nonclassical technique such as discussed later in the section on one-step mappings.

SEQUENTIAL MAPPINGS

For a complicated mapping of the form $Z = f(\zeta)$, roots are often more easily selected and the mapping is more easily created, if the process is broken up into an equivalent sequence of simpler mappings. For instance, the von Karman-Trefftz transformation of Eq. (12) can be restated as the sequence

$$\omega = \frac{Z-A}{Z-B} \quad , \tag{13a}$$

$$\eta = \omega^k \quad , \tag{13b}$$

$$\zeta = \frac{a-b\eta}{1-\eta} \quad . \tag{13c}$$

The bilinear transformations, Eqs. (13a) and (13c) require no root selection, while Eq. (13b) requires keeping track of only a single angle. Other schemes may require the separate tracking of the angles related to both the $(Z-A)^k$ and the $(Z-B)^k$ term.

The logarithmic transformation, Eq. (6), and its inverse

$$\zeta = e^Z \tag{14}$$

are the key elements of conformal mappings for a cascade of airfoils, such as occur in turbomachinery. For this case, there are an infinite number of identical airfoils in the Z plane, all of which map onto the same contour in the ζ plane. When mapping from the ζ plane to the Z plane, the correct root is the one that "tracks" a nearby point; the other roots will be different by an integral multiple of 2π in the vertical (or imaginary axis) direction. Most of the cascade mappings which the author has seen, Refs. [19], [20], [21] and [22], can be restated as a simple sequence* with Eq. (14) (possibly combined with a bilinear transformation) as the first element.

As a general principle, it is highly recommended for mappings of the form $Z = f(\zeta)$ that a complicated mapping be restated as a sequence of simpler mappings for actual implementation in a computer program.

NEAR-CIRCLE TO CIRCLE MAPPING

If an orthogonal grid is desired, the mapping of a near-circle to a circle is often chosen as one-step of the mapping process. Although there are now a plethora of ways to accomplish this mapping**, the classical technique is that of using the Theodorsen-Garrick transformation, Ref. [18], given by

---

* The author has not yet been successful in breaking up the Garrick transform, Ref. [23], into a simple form.

** Including panel techniques, Schwartz-Christoffel techniques, and elliptic techniques with orthogonal boundary control.

$$\frac{Z}{\zeta} = e^{\sum_{j=0}^{j=N}(a_j+ib_j)\zeta^j} \tag{15a}$$

or, in its derivative form by

$$\frac{dZ}{d\zeta} = e^{\sum_{j=0}^{j=N}(a_j+ib_j)\zeta^j} \tag{15b}$$

What is frequently not appreciated is that when Eqs. (15a, 15b) are implemented in the required iterative manner (see Ref. [18 or 24] for details), the iteration may not necessarily converge unless the near-circle is sufficiently "near" to being a circle. In particular, there is a requirement on the maximum value* allowed for $\left|\frac{d(\ell nr)}{d\Theta}\right|$ for the mapping in Eq. (15a), where r and Θ are the polar coordinates describing the near-circle. For the vast majority of airfoils, the classical von Karman-Trefftz transformation produces a near-circle that meets the above requirement. For turbine cascades, on the other hand, the above requirement is usually not met. It is possible to underrelax the iteration, but the required underrelaxation parameter may need to be quite small for Eq. (15a) to ensure convergence, resulting in unrealistic computation time requirements. The form given by Eq. (15b) is much less sensitive, and with an underrelaxation factor of .5 has converged even for a case where r was a multiple-valued function of Θ, so that $\left|\frac{d\ell nr}{d\Theta}\right|$ was infinite! Obviously, the convergence criterion for Eq. (15b) is substantially different than that derived for Eq. (15a) in Ref. [25]. In short, if one uses the derivative form, the chances for success are much higher. Of course, the derivative form requires the integration of a complex function, but an excellent technique using a spline exists as discussed later. Both of the above forms can employ fast Fourier techniques, as in Refs. [24, 26], to efficiently evaluate the series $\sum(a_j+ib_j)\zeta^j$ at evenly spaced increments on the unit circle in the ζ plane and thus keep the required computing time modest.

ONE-STEP MAPPINGS

A one-step mapping maps a contour onto a canonical shape, such as a circle or the real axis, in a single step. Such transformations are usually given in derivative form, namely $\frac{dZ}{d\zeta} = f(\zeta)$.

---

* For a relaxation parameter of unity, this maximum value is .2955.

The classic Schwarz-Christoffel transformation for a polygon with N straight sides takes the form

$$\frac{dZ}{d\zeta} = \prod_{j=1}^{j=N} (\zeta - b_j)^{k_j} \quad , \qquad (16)$$

while for a polygon with curved sides, Davis, Ref. [27], uses a differential form of the product term in Eq. (16) so that

$$\frac{dZ}{d\zeta} = e^{\frac{1}{\pi}\int \ell n(\zeta - b)\, d\beta} \quad , \qquad (17)$$

where β is an angle related variable. Davis employs a composite integration formula to resolve the curvature effects. Equation (16), as used by Skulsky, Ref. [28], can be considered a subset of the Davis technique. The Davis technique has been successfully employed in Refs. [12], [29], and [30] to map a wide range of complicated configurations.

An alternative is to use the form

$$\frac{dZ}{d\zeta} = g(\zeta)\, e^{\sum_{j=0}^{j=N} (a_j + ib_j)\zeta^j} \quad , \qquad (18)$$

where $g(\zeta)$ is chosen to resolve angles or general behavior*, while the exponential term accounts for the curvature. This form can take advantage of fast Fourier techniques to evaluate the coefficients a_j and b_j.

One example of such a mapping is given by

$$\frac{dZ}{d\zeta} = \left[1 - \frac{1}{\zeta}\right]^k e^{\sum_{j=0}^{j=N} (a_j + ib_j)\zeta^{-j}} \quad , \qquad (19)$$

which is used in Ref. [26] to map an airfoil in the Z plane to a circle in the ζ plane. Another mapping of this type is given by

---

* The $g(\zeta)$ function is easily constructed using the Schwarz-Christoffel technique, as illustrated in Kober, Nehari, or Milne-Thomson, Refs. [31], [32] or [33].

$$\frac{dZ}{d\zeta} = -\frac{(\zeta - iR_0)}{(1-iR_0)(\zeta+1)^2(\zeta-1)}\, e^{\sum_{j=0}^{j=N}(a_j+ib_j)\zeta^j} \quad , \qquad (20)$$

where

$$R_0 = 1 + 2\sqrt{\pi\varepsilon} \quad ,$$

and

$$\varepsilon = \frac{\text{nose radius}}{\text{inlet interior radius}} \quad .$$

This mapping was developed by the author to map the region exterior to a semi-infinite inlet and above the centerline to the interior of a circle as illustrated in Fig. 4. A similar form, suggested by Jameson, Ref. [34], given by

$$\frac{dZ}{d\zeta} = \frac{(R_0^2+\zeta^2)}{(\zeta+1)^3(1-\zeta)}\, e^{\sum_{j=0}^{j=N}(a_j+ib_j)\zeta^j} \quad , \qquad (21)$$

was used to map an inlet and centerline to a half circle as illustrated in Fig. 5.

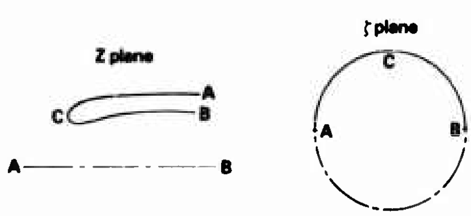

FIGURE 4. INLET TO CIRCLE MAPPING

FIGURE 5.
INLET TO HALF CIRCLE MAPPING

The latter mapping did not adequately resolve the exterior inlet contour far downstream (but was good otherwise), while a mapping based on Eq. (20) adequately resolved the inlet in all areas. The reason is that the evenly spaced point distribution (as required by fast Fourier techniques) on the ζ plane circle is too sparse when transformed back to the Z plane to accurately represent the nacelle far downstream for Eq. (21). This example illustrates the need to choose the correct canonical domain. Of course, the two canonical domains are related in a simple quadratic manner through the transformation

$$\frac{Z-1}{Z+1} = i\cdot\left[\frac{\zeta-1}{\zeta+1}\right]^2 \quad , \qquad (22)$$

which transforms a unit circle in the Z plane to a unit upper half circle (and to the real axis from -1. to 1.) in the ζ plane as illustrated in Fig. 6.

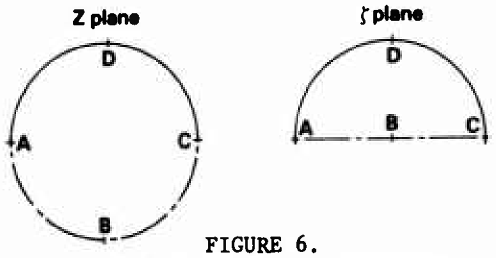

FIGURE 6.
CIRCLE TO HALF CIRCLE MAPPING

The one-step mapping of Eq. (18) is far simpler to program* than the conventional classic technique described earlier, converges stably and rapidly, and is easy to modify later for a new class of geometries. If one wishes to use fast Fourier techniques for conformal mapping, Eq. (18) is recommended as the method of choice rather than using a sequence resulting in a near-circle followed by a derivative transform like Eq. (15b), especially since Eqs. (18) and (15b) differ only by a very few lines of computer code.

It is recommended that one should not expect the Fourier series in Eqs. (15a), (15b), or (18) to resolve a slope discontinuity by the sheer brute force of a large number of Fourier terms. One might "get away" with doing so for small discontinuities in slope (e.g., 5 degrees) where a high accuracy near the discontinuity is not required, but for larger slope discontinuities the accuracy and convergence of the transformation will suffer. It is easy to remove such slope discontinuities using the hinge point transformations covered next or by incorporating slope discontinuities into the $g(\zeta)$ term in Eq. (18) so that the Fourier series only has to resolve a function with a continuous derivative.

HINGE POINT TRANSFORMATIONS

At the other extreme from one-step mappings, the hinge point transformations due to Moretti and Hall, Refs. [17] and [35], break the problem up into a large number of sequential applications of a single mapping. For problems with a plane of symmetry, Moretti applied a von Karman-Trefftz transformation to each pair of slope discontinuities in turn, to produce a smooth near-circle. Hall

---

* Note that the computer program in Ref. [26] can be easily modified to implement a mapping using Eq. (18), instead of using Eq. (19) for the airfoil.

repeatedly applied the transformation $Z = (\zeta-\zeta_0)^k$ (Eq. (10a)), where ζ_0 and k have been chosen to remove the left-most remaining slope discontinuity at each stage, to produce a smooth near half plane. Once a near-circle or near half plane is obtained, a non-conformal shearing or a conformal mapping can be used to produce a canonical domain.

MULTIPLE BODIES

Techniques to map multiple bodies can be classified as simultaneous, sequential, iterative, and periodic. This section discusses these techniques.

A simultaneous multiple body mapping maps two or more bodies simultaneously to near-canonical or canonical domains. The single mapping

$$\prod_{j=1}^{j=N} \left[\frac{\zeta-\zeta_{Tj}}{\zeta-\zeta_{Nj}}\right] = \prod_{\ell=1}^{\ell=N} \left[\frac{Z-Z_{T\ell}}{Z-Z_{N\ell}}\right]^{k_\ell} , \tag{23}$$

taken from Ref. [24] simultaneously maps N airfoils to N near-circles. The Garrick transformation in Ref. [36] simultaneously maps two concentric near-circles to two concentric circles.

A sequential multiple body mapping maps two contours to canonical contours by first mapping one contour to a canonical contour, then mapping the second contour to a canonical contour while preserving the nature of the first canonical contour. Two examples of this are given in Ref. [24], where transformations are given to map an airfoil to a near-circle (or to map a near-circle to a circle) while keeping a nearby circle a circle. Another example is illustrated in Fig. 7, where a centerbody is mapped onto the real axis using a technique similar to that in Ref. [37], and then a nearby inlet is mapped to a circle, using the Eq. (20), while keeping the real axis a straight line.

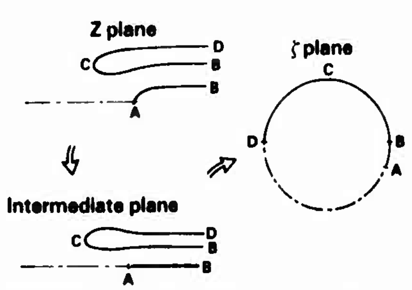

FIGURE 7. INLET/CENTERBODY TO CIRCLE MAPPING SEQUENCE

After doing a number of problems both simultaneously and sequentially, the author has found that a sequential multiple body technique usually works better in practice than a simultaneous technique since the branch cuts required to resolve the root selection problem are simpler. This more than balances the increase in computer time* required for a sequential approach.

An iterative technique to map multiple contours to canonical contours has recently been developed by Halsey, Ref. [38]. In this technique each contour is individually mapped to a circle in sequence, with no special constraints on holding the other contour shapes. By repeatedly cycling through all the contours one at a time, the contours are all mapped to circles to engineering accuracy. That this process converges is not particularly surprising; what is surprising is that only about five cycles through all the contours are required to achieve four digit accuracy for practical cases.

A periodic mapping maps a finite or infinite number of identical (but otherwise displaced or rotated) contours onto overlayed (but otherwise identical) contours on different Riemann sheets. As an example, the logarithmic transformation of Eq. (6), when appropriately scaled, maps a cascade of airfoils in the ω plane onto a single (highly distorted) airfoil-like contour in the ζ plane. The transformation $Z = \zeta^N$ (which is a subset of Eq. (10a)), where N is an integer, maps a region with N angular periodic boundaries in the Z plane onto a region with one periodic boundary in the ζ plane. By these means, a periodic configuration can be reduced to a form wherein the periodicity condition reduces essentially to a continuity condition (except for circulation-type terms associated with branch cuts).

CREATING NEW MAPPINGS

This section includes a discussion of a few general techniques that can be used to create new conformal mappings, and some restrictions to keep in mind while doing so.

The first step when considering the development of a new mapping is to verify that the desired mapping has not already been created. The book by Kober, Ref. [31], includes an extensive survey of mappings developed up to the mid 1940's, and should be reviewed first. The reference material in Table 1, included later in this paper, can be consulted by topic. If none of the above include the desired mapping, a careful literature search may be warranted. Only when the investigator is reasonably sure of not "reinventing the wheel" should a new mapping be attempted.

* A 2-body sequential approach may typically require twice the computer time of a simultaneous approach.

If the complex potential ω for the two-dimensional incompressible flow over (or through) a body is known, i.e., if

$$\omega = \phi + i\psi = f(Z)$$

is known, then the mapping from the Z plane to the ω plane (with coordinates (ϕ,ψ)) is known. Thus, knowing the incompressible flow is equivalent to knowing the mapping from the Z plane to a rectangle in the ω plane, and vice versa. This principle can be used to construct conformal mappings. In particular, a "panel method" potential flow program can be used to construct a conformal mapping, Ref. [39].

Another technique for the creation of new mappings can be summarized by the description "guess and plot." Using this technique a function having appropriate zeroes, poles, and singularities is guessed*, and its effect is determined by plotting a transformed contour or a transformed grid. About one third of the functions this author has guessed over the years have had the appropriate action, indicating that this technique is more viable than might be thought at first glance.

Since a conformal mapping is simply a functional relationship, if one knows

$$Z = f(\omega) \quad ,$$

and

$$\zeta = g(\omega) \quad ,$$

then one has

$$Z = f(g^{-1}(\zeta)) \quad .$$

In short, if one knows how to map two different contours to the same contour, then one can map one of these contours onto the other, Refs. [31] and [40]. Put another way, a conformal mapping of a conformal mapping is a conformal mapping. One then begins to think in terms of building up a sequence of mappings, with each step bringing a contour closer to a canonical contour such as a unit circle. Thus by combining known mappings in a new sequence, one can construct new mappings.

In generating an orthogonal grid with conformal mapping, there are two processes involved. The first process is to conformally map the desired contour (i.e: an airfoil, or inlet, etc. ...) to a rectangle (or equivalently to a circle or half plane). The computing time required for this step is often

---

* These singularities, poles, and zeroes are placed at points on the contour where the slope is discontinuous, and when necessary (e.g., near regions of high convex curvature of the contour), inside the contour along a line joining the local contour point of maximum curvature and its center of curvature. How far along this line depends on the total slope change nearby, and the singularity powers also depend on these slope changes.

nearly lineariy* proportional to L, the number of input points. The second process is classically to map an orthogonal rectangular grid in the mapped plane back to the physical plane. This involves a computing time proportional to MxN, the total number of grid points. It is common for MxN to be substantially larger than L. This implies that the forward transformations can be somewhat inefficient (i.e., implicit) but the inverse transformations should be efficient (i.e., explicit), if possible. This should be kept in mind when creating mappings that are to be used to transform a grid back to the physical plane.

When creating new mappings, one principle to be followed is "do not create singularities** within the flow field"; however, singularities are often necessary and completely acceptable within a body, on a bounding surface, or at images of "infinity." An example of creating a singularity within the flow field is shown in Fig. 8, taken from Ref. [19]. Figure 9 from this same reference shows a grid system with the same number of grid points which is better suited for flow computations. The above principle was used in Ref. [24] to determine some mapping parameters that were otherwise undetermined. With another choice of parameters, the mappings therein looked something like those in Fig. 10 rather than like those in Fig. 11 where the parameters were chosen to obey this principle.

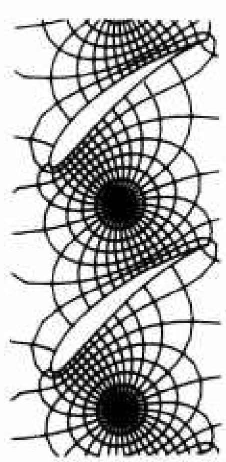

FIGURE 8. GRID SYSTEM WITH SINGULARITY IN FLOW FIELD

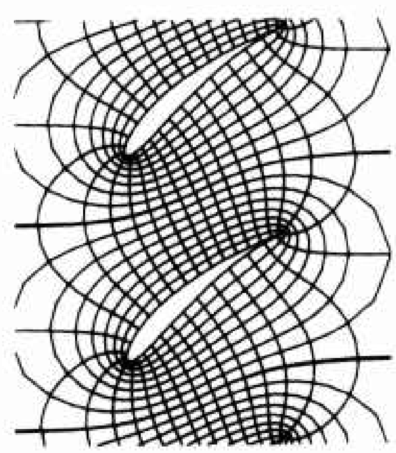

FIGURE 9. GRID SYSTEM WITHOUT SINGULARITY IN FLOW FIELD

* Actually the operation count goes as $L \times \ell n_2 L$, as dictated by using fast Fourier techniques for Eqs. (15a), (15b), or (18).

** A singularity occurs when $\frac{dZ}{d\zeta} = 0$ or infinity.

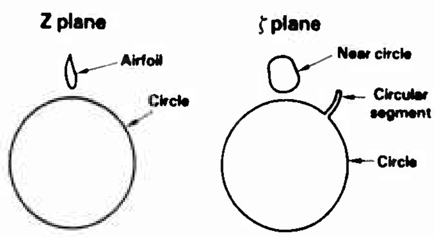

FIGURE 10. INCORRECT CHOICE OF MAPPING CONSTANTS

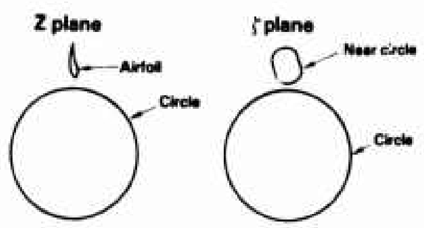

FIGURE 11. CORRECT CHOICE OF MAPPING CONSTANTS

REFLECTION PRINCIPLE

It is possible to map a contour to a desired shape while simultaneously keeping a nearby straight line straight by using a reflection principle. If the relation

$$f(Z) = g(\zeta)$$

conformally maps a contour from the Z plane to the ζ plane in a desirable manner, then the relations

$$f(Z)\cdot\bar{f}(Z) = g(\zeta)\cdot\bar{g}(\zeta) \quad , \tag{24a}$$

and

$$\frac{f(Z)}{\bar{f}(Z)} = \frac{g(\zeta)}{\bar{g}(\zeta)} \quad , \tag{24b}$$

where the superscript "-" denotes the complex conjugate operator, will both have a "similar"* effect but will preserve the shape of the real axis. A proof of the real axis shape preservation is outlined in Appendix A. Examples involving mappings symmetric with respect to a line appear in Refs. [16] and [17]. An extension of the above concept allows the construction of mappings which preserve a circle by utilizing operators involving image points with respect to circles rather than the complex conjugate operator. Examples of circle preserving mappings are contained in Refs. [24] and [36]. The question of whether to use the product mapping, Eq. (24a), or the ratio mapping, Eq. (24b), can be resolved by simply programming both and then choosing the one which produces the most desirable effect.

* "Similar" is taken here to mean not identical, but generally of the same nature.

CALCULATING THE GRID

Recently, Sockol and Adamczyk in Refs. [39] and [41] have observed that the generation of an orthogonal grid can be broken down into two independent steps, namely;

1) calculation of the boundary point correspondence between the physical and computational planes by means of conformal mapping or some other orthogonality preserving technique, and
2) the generation of a grid given this boundary point correspondence.

This observation is to conformal grid generation as the finite volume (or finite element) technique is to flow calculations; it breaks the grid generation problem up into two nearby independent steps just as the finite volume technique breaks a flow simulation up into two nearby independent steps (namely the generation of a grid and the solution of the flow equations on the grid). This allows a great deal of flexibility to be attained. In short, what is the most efficient, simple, or general technique required to complete step #1 (which is basically a boundary operation) is not necessarily the same as the most efficient, simple, or general technique required to complete step #2 (which is basically an area operation). This represents a major step forward in orthogonal grid generation. This section reviews some methods used to complete step #2.

One obvious way to generate an orthogonal grid is to invert the conformal mapping used in step #1, and then map an orthogonal grid constructed in the computational plane back to the physical plane, as mentioned earlier. If one has used an algebraic stretching in the direction tangential to the body to create the computational grid, fast Fourier techniques cannot be easily employed in the mapping inversion. Depending on the mapping complexity, the number of Fourier terms, and the presence of implicit expressions in the inverse mapping, the creation of a grid at NxM points can require the order of CxLxNxM operations, where L is the number of Fourier terms and C is a relatively large constant. In practical terms, this can require about six seconds of time on an IBM 3081 computer for a typical* case. For a non-stretched isolated airfoil mapping, where fast Fourier techniques can be used in the inverse mapping, a typical grid can be generated in about two seconds of IBM 3081 computer time, once the conformal mapping is known.

---

* For the purposes of this paper "typical" means a 128x32 grid using 256 Fourier terms for an inlet/centerbody configuration, unless otherwise specified. The IBM 3081 computing time quoted can be taken as roughly equivalent to an equal amount of CDC 7600 computer time.

Another way to generate an orthogonal grid is to solve a Laplace problem (with Dirichlet boundary conditions) on the computational grid for both x and y (the physical plane coordinates), which are known on the boundary as described in Refs. [39] and [41]. Since the Laplacian operator will retain orthogonality, the (x,y) values thus calculated will describe an orthogonal grid in the physical plane. A fast Poisson solver, Ref. [42], with an operation count of $C \times N \times M \times \ell n_2 N$, where C is a relatively small constant, may be used to solve the Laplace problem. For a typical grid, this will require about two seconds on an IBM 3081 computer. This solver is sufficiently general even for use when independent stretchings are used following a conformal mapping to generate the rectangular grid in the computational plane.

THREE DIMENSIONS

Conformal mapping is basically a surface technique, but it can be used as one component of a three-dimensional grid generation system. One example is shown in Fig. 12, taken from Ref. [16], where a three-dimensional inlet/centerbody grid was constructed using conformal mapping in each circumferential plane, or slice, followed by independent algebraic stretchings to construct an orthogonal grid in the sliced plane, one of which is shown. The resultant three-dimensional grid is orthogonal in two directions, but not in the third. Similar techniques are used to construct near-orthogonal grids in Refs. [13], [43], and [44].

REFERENCE MATERIAL

The following table summarizes a limited amount of reference material for conformal mapping of aerodynamic configurations. The book by Kober, Ref. [31], is particularly useful when attempting to map a new configuration. The paper by Moretti, Ref. [45], is highly recommended as an alternate review of the field.

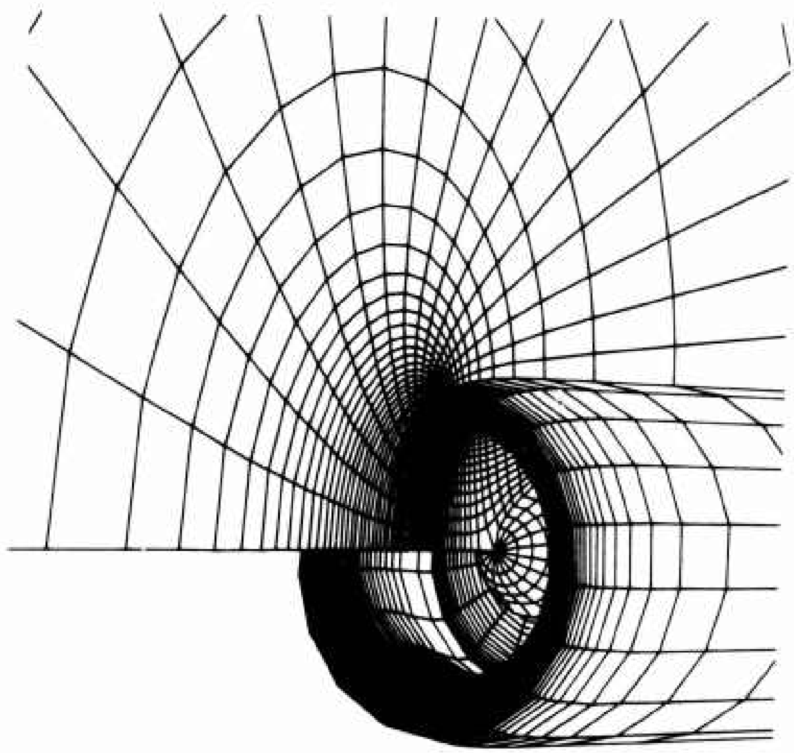

FIGURE 12. PERSPECTIVE VIEW OF
THREE-DIMENSIONAL COORDINATE SYSTEM

TABLE 1 - A LIMITED SET OF CONFORMAL MAPPING REFERENCES.

Subject	References
General - Papers	27, 28, 40, 45, 46, 47, 48
General - Books	11, 31, 32, 33
Airfoil	18, 24, 26
Airfoil with Spoiler	49
Two Piece Airfoil	24, 36, 38
Multiple Bodies	24, 38
Inlet	16, 50, 51
Inlet with Centerbody	16
Axisymmetric Body	37
Cascade	19, 20, 21, 22, 23, 39, 41
Non-Planar	4, 5, 6, 7, 8, 9, 10, 12, 13, 14
Symmetric	16, 17
Ducts	27, 29, 30

PACKAGED TOOLS

To accomplish simple conformal transformations, one does not need much in the way of supporting tools. For more complex transformations, a number of packaged tools can substantially simplify program development. Among these tools are spline fits, fast Fourier transforms, efficient matrix techniques, fast Poisson solvers, near-circle to circle transformation packages, and good graphics. The need for, use, and availability of these packages is covered in this section.

Mapping procedures often require interpolation between points. A spline fit is a good way to accomplish such an interpolation, Ref. [52]. In addition, it may be necessary to numerically integrate a function. Such an integration can be accomplished by analytic integration of a spline passed through the function values. A spline routine accomplishing both of these objectives is listed in a FORTRAN program form on pages 279-281 of Ref. [26]. By simply declaring all floating point variables in this routine as COMPLEX, this program is suitable for integrating a complex function. The derivative form of transformations of Eqs. (15b), (16), (17), and (18) require such an integration.

A fast Fourier transform, such as given in Ref. [26], is a key element of an efficient implementation of transformations using Eqs. (15a), (15b), or (18). The program listing contained within pages 202 to 240 of Ref. [26] uses fast Fourier techniques, and makes a good starting point.

The LINPACK package available from the Society for Industrial and Applied Mathematics (SIAM), Ref. [53], contains linear equation solvers that can be useful in mapping operations. If one uses a near-circle to circle transform at unevenly spaced points in the circle plane (namely at points corresponding to the input points in the physical plane), fast Fourier techniques cannot be used. By using LINPACK routines on a fast computer (i.e: IBM 3081) moderate computation times (about one minute of CPU time) are required for a representation involving one or two hundred points. A typical FFT mapping time for the same number of points is two seconds.

Fast Poisson solvers are available for filling-in the grid, as mentioned previously. A rather versatile fast Poisson solver is reported in Ref. [42].

Good graphics routines for plotting intermediate contours and grids have, in practice, been found to be the most important element of the process of inventing new mappings, or new combinations of mappings. By using extensive graphics, a defective mapping is soon discovered with little effort. This allows a wider range of functions to be tried (guessed) in the given amount of time allowed for completion of a project. The graphics package must be general enough to plot both contours and grids and should be accessible by a simple subroutine call. It is not unusual to plot fifty different intermediate results during the development of a mapping sequence such as that in Ref. [16]. Of course, each plotting call is sequentially commented out, but not deleted, during the development process. Such an extensive use of graphics greatly simplifies debugging, but would be unbearably tedious without a simple calling sequence. Since graphics tends to be tailored to a particular computer installation, a sufficiently versatile and easy to use package often must be written locally and thus is not usually available to the public.

CLOSING REMARKS

Modern finite volume flow solvers do not require an orthogonal grid, but a near-orthogonal grid is usually beneficial. Often a simple conformal transformation, followed by independent algebraic transformations, can be used to generate such a near-orthogonal grid with little difficulty to be expected in a computer program implementation. This is a good way to get involved with conformal mapping.

The conformal mapping of a contour onto a canonical contour is far easier to accomplish using a one-step technique based on Eq. (18), as opposed to classically mapping a contour onto a near-circle and then mapping the near-circle onto a circle. This is especially true since Ref. [26] contains a one-step conformal mapping computer program which is easily modified to map new geometries. This one-step mapping technique is faster, simpler, more resistant to root selection problems, and more stable than the classical technique.

It is not necessary to generate the grid using the inverse of the transform employed to map the contour to a canonical contour. In fact, it would seem that use of a fast Poisson solver to generate the grid using the known boundary correspondence offers a flexibility, simplicity, and economy that may not be surpassed by other methods.

REFERENCES

1. Thompson, J.F. and Mastin, C.W. (1980) "Grid Generation Using Differential Systems Techniques," NASA Conference Publication No. 2166 on Numerical Grid Generation Techniques, pp. 37-72, Hampton, Va.
2. McFadden, G.B. (1979) "An Artificial Viscosity Method for the Design of Supercritical Airfoils," Report COO-3077-158, Courant Mathematics and Computing Laboratory, New York University, N.Y., N.Y.
3. Volpe, G. and Melnik, R.E. (1981) "The Role of Constraints in the Inverse Design Problem for Transonic Airfoils," Paper No. 81-1233, AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, California.
4. Young, L. (1958) "Runners of Experimental Turbomachines," Engineering, pp. 376-378.

REFERENCES (continued)

5. Prasil, F. (1926) "Techniche Hydrodynamik," Appendix 4, J. Springer, Berlin.
6. Senoo, Y. and Yoshiyuki, N. (1971) "A Blade Theory of an Impeller with an Arbitrary Surface of Revolution," Trans. ASME Journal of Engineering for Power, pp. 454-460.
7. Wislicenus, G. (1947) "Fluid Mechanics of Turbomachinery," McGraw-Hill, New York.
8. Lewis, R.A. (1964-1965) "Internal Aerodynamics of Turbo-Machines," Proc. Instn. Mech. Engrs., Vol. 179, Pt. 1, pp. 1115-1128.
9. Weatherburn, C.E. (1955) "Differential Geometry of Three-Dimensions," Volume 1, Cambridge at the University Press.
10. Kumar, T.C.M. and Rao, Y.V.N. (1977) "Theoretical Investigation of Pressure Distributions Along the Surfaces of a Thin Blade of Arbitrary Geometry of a Two-Dimensional Centrifugal Pump Impeller," ASME Journal of Fluids Engineering, pp. 531-542.
11. Polya, G. and Latta, G. (1974) "Complex Variables," John Wiley & Sons, New York, N.Y.
12. Grossman, B. (1979) "Numerical Procedure for the Computation of Irrotational Conical Flows," AIAA Journal, Vol. 17, No. 8, Article No. 78-1213R, pp. 828-837.
13. Grossman, B. and Siclari, M.J. (1980) "The Non-Linear Supersonic Potential Flow Over Delta Wings," Paper No. 80-0269, AIAA 18th Aerospace Sciences Meeting, Pasadena, California.
14. Murata, S., Miyake, Y. and Kawabata, N. (1980) "Exact Two-Dimensional Theory of Spherical Spiral Groove Bearings," ASME Journal of Lubrication Technology, Vol. 102, pp. 430-438.
15. von Karman, T. and Trefftz, E. (1918) "Potential-stromung um gegebene Tragflachenquerschnitte," Zeitschrift fur Flugtechnische Moforluftsch, Vol. 9, pp. 111-116.
16. Ives, D.C. and Menor, W.A. (1981) "Grid Generation for Inlet and Inlet-Centerbody Configurations Using Conformal Mapping and Stretching," Paper No. 81-0997, AIAA 5th Computational Fluid Dynamics Conference, Palo Alto, Calif.
17. Moretti, G. (1976) "Conformal Mappings for Computations of Steady, Three-Dimensional, Supersonic Flows," Numerical/Laboratory Computer Methods in Fluid Mechanics, ASME.
18. Theodorsen, T. and Garrick, I.E. (1933) "General Potential Theory of Arbitrary Wing Sections," NACA TR 452.
19. Ives, D.C. and Liutermoza, J.F. (1977) "Analysis of Transonic Cascade Flow Using Conformal Mapping and Relaxation Techniques," AIAA Journal, Vol. 15, pp. 647-652.
20. Howell, A.R. (1948) "A Theory of Arbitrary Airfoils in Cascade," The Philosophical Magazine, Vol. 39, pp. 913-927.
21. Legendre, R. (1972) "Work in Progress in France Related to Computation of Profiles for Turbomachine Blades by Hodograph Method," American Society of Mechanical Engineers, Paper 72-GT-41.
22. Frith, D.A. (1973) "Inviscid Flow Through a Cascade of Thick, Cambered Airfoils, Part 1 - Incompressible Flow," American Society of Mechanical Engineers, Paper 73-GT-84.
23. Garrick, I.E. (1944) "On the Plane Potential Flow Past a Lattice of Arbitrary Airfoils," NACA Rept. 688.
24. Ives, D.C. (1976) "A Modern Look at Conformal Mapping, Including Multiply Connected Regions," AIAA Journal, Vol. 14, pp. 1006-1011.
25. Warschawski, S.E. (1945) "On Theodorsen's Method of Conformal Mapping of Nearly Circular Regions," Quarterly of Applied Mathematics, Vol. 3, pp. 12-28.

REFERENCES (continued)

26. Bauer, F., Garabedian, P., Korn, D. and Jameson, A. (1975) "Supercritical Wing Sections II," Vol. 108, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, New York.
27. Davis, R.T. (1979) "Numerical Methods for Coordinate Generation Based on Schwarz-Christoffel Transformation," Paper No. 79-1463, AIAA 4th Computational Fluid Dynamics Conference, Williamsburg, Va.
28. Skulsky, R.S. (1966) "A Conformal Mapping Method to Predict Low-Speed Aerodynami Characteristics of Arbitrary Slender Re-entry Shapes," AIAA Journal of Spacecraft, Vol. 3, pp. 247-253.
29. Sridhar, K.P. and Davis, R.T. (1981) "A Schwarz-Christoffel Method for Generating Internal Flow Grids," Symposium on Computers in Flow Prediction and Fluid Dynamic Experiments, pp. 35-44, ASME Winter Annual Meeting, Washington, D.C.
30. Anderson, O.L., et al (1982) "Solution of Viscous Internal Flows on Curvilinear Grids Generated by Schwarz-Christoffel Transformation," Symposium on the Numerical Generation of Curvilinear Coordinate Systems and use in the Numerical Solution of Partial Differential Equations, Nashville, Tenn.
31. Kober, H. (1957) "Dictionary of Conformal Representations," Dover Publications, Inc., New York.
32. Nehari, Z. (1952) "Conformal Mapping," Dover Publications, Inc., New York.
33. Milne-Thomson, L.E. (1960) "Theoretical Hydrodynamics," The MacMillan Company, New York, N.Y.
34. Jameson, A. (1979) Private Communication.
35. Hall, D.W. (1980) "A Three-Dimensional Body-Fitted Coordinate System for Flow Field Calculations on Asymmetric Nosetips," NASA Conference Publication No. 2166 on Numerical Grid Generation Techniques, pp. 315-328, Hampton, Va.
36. Garrick, I.E. (1936) "Potential Flow About Arbitrary Biplane Wing Sections," Rept. 542, NACA.
37. Arlinger, B. (1980) "Axisymmetric Transonic Flow Computations Using a Multigrid Method," Proceedings of Seventh International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, Vol. 141, pp. 55-60, Springer-Verlag, New York, N.Y.
38. Halsey, N.D. (1979) "Potential Flow Analysis of Multielement Airfoils Using Conformal Mapping," AIAA Journal, Vol. 17, No. 12, pp. 1281-1288.
39. Adamczyk, J.J. (1980) "2D Grid Code for Rotor and Stator Blade Sections Using Electrostatic Analogy - O-Type and C-Type," NASA Conference Publication No. 2165 on Numerical Grid Generation Techniques, pp. 129-142, Hampton, Va.
40. Caughey, D. (1978) "A Systematic Procedure for Generating Useful Conformal Mappings, Int. J. Numerical Methods in Eng., 12, p. 1651.
41. Sockol, P.M. (1980) "2D Grid Code for Rotor and Stator Blade Sections - C-Type," NASA Conference Publication No. 2166 on Numerical Grid Generation Techniques, pp. 437-448, Hampton, Va.
42. Swarztrauber, P.N. and Sweet, R. (1975) "Efficient Fortran Subprograms for the Solution of Elliptic Partial Differential Equations," Tech. Note NCAR-TN/IA-109, National Center for Atmospheric Research, Boulder, Colorado.
43. Jameson, A. (1974) "Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1," Communications on Pure and Applied Mathematics, Vol. XXVII, pp. 283-309.
44. Jameson, A. and Caughey, D.A. (1977) "A Finite Volume Method for Transonic Potential Flow Calculations," Paper No. 77-635, AIAA 3rd Computational Fluid Dynamics Conference.

REFERENCES (continued)

45. Moretti, G. (1980) "Grid Generation Using Classical Techniques," NASA Conference Publication No. 2166 on Numerical Grid Generation Techniques, pp. 1-35, Hampton, Va.
46. Laura, P.A.A., (1975) "A Survey of Modern Applications of the Method of Conformal Mapping," Revista de la Union Mathematica Argentina, Vol. 27, pp. 167-179.
47. Garrick, I.E. (1949) "Conformal Mapping in Aerodynamics, with Emphasis on the Method of Successive Conjugates," Symposium on Construction and Applications of Conformal Maps, National Bureau of Standards, Applied Mathematics Series, Vol. 18, pp. 137-147.
48. Smith, R.E. (1980) ed., Numerical Grid Generation Techniques, NASA Conference Publication No. 2166, Hampton, Va.
49. Rossow, V.J. (1973) "Conformal Mapping for Potential Flow About Airfoils with Attached Flap," AIAA Journal of Aircraft, Vol. 10, pp. 60-62.
50. Arlinger, B.G. (1975) "Calculation of Transonic Flow Around Axisymmetric Inlets," Paper No. 75-80, AIAA 13th Aerospace Sciences Meeting, Pasadena, California.
51. Caughey, D.A. and Jameson, A. (1976) "Accelerated Iterative Calculation of Transonic Nacelle Flowfields," AIAA Paper 76-100, Washington, D.C.
52. Ahlberg, J.H., Nilson, E.N. and Walsh, J.L. (1967) The Theory of Splines and Their Applications, Academic Press, New York, pp. 9-16.
53. Dongarra, J.J. (1979) "LINPACK User's Guide," Society of Industrial and Applied Mathematics, Philadelphia, Pa.

APPENDIX A

PRODUCT MAPPING

Consider the product mapping

$$f(Z)\cdot\bar{f}(Z) = g(\zeta)\cdot\bar{g}(\zeta) = \omega = \phi + i\psi \quad .$$

The real axis in the Z plane can be specified by the relation $Z = \bar{Z}$, so on the real axis

$$\begin{aligned} f(Z)\cdot\bar{f}(Z) &= f(Z)\cdot\bar{f}(\bar{Z}) \\ &= f(Z)\cdot\overline{f(Z)} \\ &= \text{pure real} = \phi + i\psi \quad . \end{aligned}$$

Therefore, the real axis $Z = \bar{Z}$ maps to the real axis $\psi = 0$ in the ω-plane. In a similar manner, the real axis $\zeta = \bar{\zeta}$ also maps to the real axis $\psi = 0$ in the ω plane. Thus the real axis $Z = \bar{Z}$ maps to the real axis $\zeta = \bar{\zeta}$.

RATIO MAPPING

Consider the ratio mapping

$$\frac{f(Z)}{\bar{f}(Z)} = \frac{g(\zeta)}{\bar{g}(Z)} = e^{\omega} = e^{\phi+i\psi} \quad .$$

APPENDIX A (continued)

The real axis in the Z plane is specified by $Z = \bar{Z}$, so on the real axis,

$$\ell n \frac{f(Z)}{\bar{f}(Z)} = \ell n \frac{f(Z)}{\bar{f}(\bar{Z})}$$

$$= \ell n \frac{f(Z)}{\overline{f(Z)}}$$

$$= \text{pure imaginary} = \phi + i\psi \quad .$$

Therefore, the real axis $Z = \bar{Z}$ maps to the imaginary axis $\phi = 0$ in the ω plane. In a similar manner, the real axis $\zeta = \bar{\zeta}$ also maps to the imaginary axis $\phi = 0$ in the ω plane. Thus the real axis $Z = \bar{Z}$ maps to the real axis $\zeta = \bar{\zeta}$.